CHALMERS
UNIVERSITY OF TECHNOLOGY

Collision-Aware Communication for Intersection Management of Automated Vehicles

Citation for the original published paper (version of record):
Steinmetz, E., Hult, R., Zou, Z. et al (2018). Collision-Aware Communication for Intersection Management of Automated Vehicles. IEEE Access, 6: 77359-77371.
http://dx.doi.org/10.1109/ACCESS.2018.2882607

Collision-Aware Communication for Intersection Management of Automated Vehicles

Erik Steinmetz, Robert Hult, Zhenhua Zou, Ragne Emardson,
Fredrik Brännström, Paolo Falcone, and Henk Wymeersch

***Abstract*—Intersection management of automated vehicles relies on wireless communication, whereby communication resources should be allocated to vehicles while maintaining safety. We present a collision-aware resource allocation (CARA) strategy for coordination of automated and connected vehicles by a centralized intersection manager. The proposed strategy is based on a self-triggered approach and proactively reduces the risk of channel congestion by only assigning communication resources to vehicles that are in critical configurations, i.e., when there is a risk for a future collision. Compared to collision-agnostic communication strategies, typically considered for automated intersection management, the CARA strategy aims to bridge the gap between control, sensing and communication. It is shown to significantly reduce the required amount of communication (albeit with a slight increase in the control cost), without compromising safety. Furthermore, control cost can be reduced by allowing more frequent communication, which we demonstrate through a trade-off analysis between control performance and communication load. Hence, CARA can operate in communication-limited scenarios, but also be modified for scenarios where the control cost is of primary interest.**

I. Introduction

One of the most pressing issues in road transport systems is safe and efficient coordination of vehicles in traffic zones where roads cross or merge, such as intersections, on-ramps and roundabouts. Automated intersection management methods promise to provide safe and efficient intersection crossing, and have been the subject of recent research [1]–[4]. Such methods rely on a central controller, which periodically receives state information (ranging from position and velocity to HD video) from each vehicle, and then issues control commands which minimize a measure of cost (e.g., fuel consumption) while allowing safe passage. However, automated intersection management approaches have mainly focused on different control and coordination methods while few have specifically looked at how to optimize the use of the wireless spectrum for these coordination scenarios. This is especially important when state information requires significant amounts of data. In general, wireless links in networked control systems are affected by (i) limited bandwidth; (ii) delays; (iii) packet dropouts [5]. Optimal use of the spectrum would thus be beneficial to reduce the channel load and efficiently use the available bandwidth. Furthermore, it would make it possible to support higher densities of vehicles compared to a fixed schedule. As the methods from [1]–[4] largely considers the control, sensing and communication systems as separate entities that provide services to each other without much mutual knowledge, there are limited options to reduce the channel load. However, when communication is aware of the control entity [6], communication losses and information losses can be tolerated to some extent. This leads to the problem of optimally assigning communication resources, without compromising safety and severely affecting control performance.

The type of problem described above is closely related to resource allocation and state-based scheduling for networked control systems, which have been studied extensively, see e.g., [7]–[17]. Out of these works, the majority [7]–[12], [14]–[16] focus on the communication between the sensor and the controller, while [13] considers the communication between the controller and the actuator. We can further group these works based on the mechanisms behind the resource allocation. In [7], [8], [11]–[14], [16], communication resources are assigned based on self- or event-based triggers, and agents only transmits when specific triggering conditions are met. The works [9], [15], [17], on the other hand, focus on the problem of how to optimally schedule a group of agents over a limited number of channels, i.e., how to optimally use the available communication resources in each time slot. Another important consideration is whether the decision to transmit is made in a distributed fashion (i.e., locally by each agent) or by a central unit, and what information the decision to transmit is based on. The works [7], [9], [10], [14], [16] consider distributed protocols while [12], [15], [17] consider centralized solutions. Regarding what the decision is based on, the majority of the works rely on solutions where observations of the state are continuously available and needed for the decision of when to transmit. Only few works [15]–[17] consider the case relevant to our setting, when observations are partially available or not available at all, and the decision has to be based on predictions of the state. Out of these, [15] rely on a concept referred to as cost of information loss (CoIL)

E. Steinmetz is with the Division of Safety and Transport at RISE Research Institutes of Sweden, Borås Sweden, (e-mail: erik.steinmetz@ri.se). E. Steinmetz is also with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. R. Hult, F. Brännström, P. Falcone, and H. Wymeersch are with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden, e-mails: {robert.hult, fredrik.brannstrom, paolo.falcone, henkw}@chalmers.se. R. Emardson is with University of Borås, Borås, Sweden (e-mail: ragne.emardson@hb.se). Z. Zou is with Ericsson Research, Stockholm, Sweden (e-mail: zhenhua.zou@ericsson.com). This research was supported, in part, by the National Metrology Institute hosted at RISE Research Institutes of Sweden, which in turn is partly funded by VINNOVA under the program for national metrology (project 2015-06478); the PRoPART (Precise and Robust Positioning for Automated Road Transport) project, funded by the European GNSS Agency under the EU H2020 program (grant agreement No 776307); and the COPPLAR (CampusShuttle Cooperative Perception and Planning Platform) project, funded under Strategic Vehicle Research and Innovation grant 2015-04849.

to determine which group of agents that should transmit in each time slot, while the resource allocation in [17] is based on the principle of maximum predicted error first (MPEF). In contrast, [16] focus on a distributed setting where agents makes promises to each other regarding their intentions. These promises, which can be anything from loose descriptions of reachability sets to tight state trajectories are then used in an event- and self-triggered fashion to determine when agents should request and send updated information to each other. Besides this, [14] studies the problem of designing a self-triggered communication scheme for an encoder/decoder pair while ensuring stochastic stability of a vehicular system.

In this paper, we aim to understand if it is possible to develop a communication strategy for safe intersection management, in order to minimize the communication resources. We achieve this through a self-triggered approach [16], specialized for a safety-critical scenario, in which vehicles send state information in the uplink and a controller issues commands in the downlink. Our specific contributions are

- A novel optimization formulation for uplink scheduling over a time horizon for remote intersection management, compatible with standard controllers (e.g., model predictive control).
- A novel efficient method to characterize the possibilities of collision in the presence of state uncertainties.
- A collision-aware resource allocation (CARA) strategy, which is based on the possibilities of collision, and takes into account the coupling in the dynamics between vehicles (due to safety constraints) by assigning communication resource in a receding horizon fashion.

The proposed CARA strategy is evaluated for a two-vehicle scenario and is demonstrated to lead to significant reductions in communication load, without compromising safety, though at an increased control cost.

The rest of the paper is organized as follows. Section II, describes the particular scenario that we consider in detail and introduces the system model. In Section III, we characterize the pairwise possibilities of collision between vehicles required for the resource allocation. Section IV, details the proposed resource allocation procedure, as well as a brief example to provide some intuition. In Section V, we perform a more in depth analysis of the performance of the proposed resource allocation procedure for a simplified two vehicle scenario. Finally, in Section VI we conclude and summarize the paper, and discuss directions for future work.

Notation: In this paper matrices are denoted by uppercase bold letters, e.g., $\mathbf{X}$, vectors are denoted by lowercase bold letters, e.g., $\mathbf{x}$, and sets by calligraphic letters. e.g., $\mathcal{X}$. Furthermore, to separate between vectors and intervals we write row vectors as $[a\ b]$ while intervals are written as $[a, b]$. The transpose of a matrix $\mathbf{A}$ is denoted by $\mathbf{A}^{\mathrm{T}}$, $\mathrm{Conv}(\cdot)$ is the convex hull operator, and the set of positive real numbers is denoted by $\mathbb{R}^{+}$.

II. System Model

We consider an intersection and N vehicles. The intersection is operated by an intersection manager (IM), comprising a traffic controller, used to orchestrate the flow of the vehicles, and a resource allocator, used to assign communication resources to vehicles. As our focus is on the scheduling of the communication to reduce the overall communication load, we will consider the wireless channels to be error-free and with negligible delay. This assumption is usually implicit in contributions focusing solely on the control side of the intersection problem (e.g., [18]–[20]), and is in fact not so far fetched as upcoming 5G networks aim at providing ultra-reliable and low latency services [21].

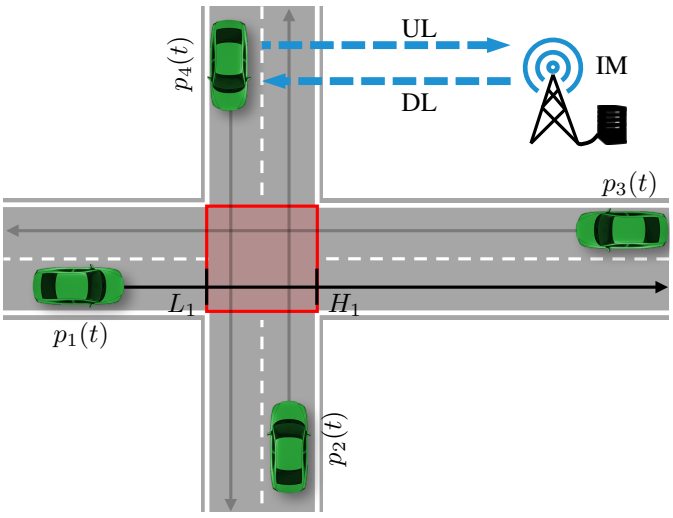


Figure 1. Schematic illustration of the intersection with definitions. The black lines illustrates the path of the vehicles and the red square is the area where collisions between vehicles on different roads can occur.

A. Vehicles and Intersection

We assume that the vehicles move along predefined and fixed paths and that their motion therefore can be considered one-dimensional, as illustrated in Figure 1. This is a standard assumption for autonomous vehicles moving in a structured environment, like an urban area [19], [22], [23]. As a matter of fact, in urban scenarios collisions with other vehicles and pedestrians are likely to be avoided by braking intervention, to a larger extent then steering, as lateral vehicle manoeuvres may increase the risk of collision with oncoming traffic in the opposite direction.

Furthermore, we model the vehicles as point-masses with state $\mathbf{x}_i(t) = [p_i(t)\ v_i(t)]^{\mathrm{T}}$, where $p_i(t)$ and $v_i(t)$ represent the scalar position and velocity of vehicle i along its fixed path, respectively. In particular, we model the vehicle motion using a stochastic linear differential equation

$$\dot{\mathbf{x}}_i(t) \quad = \quad \mathbf{A}_i\mathbf{x}_i(t) + \mathbf{b}_i u_i(t) + \mathbf{w}_i(t), \tag{1}$$

where $u_i(t) \in [u_{i,\min}, u_{i,\max}]$ is the control input, $\mathbf{A}_i$ and $\mathbf{b}_i$ are known and of appropriate dimensions, and $\mathbf{w}_i(t)$ is a stochastic perturbation. For technical reasons we also assume that the velocity is non negative, i.e, $v_i(t) \geq 0\ \forall t$. Note that this implies that the vehicles can not reverse.

As illustrated in Figure 1, we represent the intersection as intervals on the path of each vehicle, with lower and

upper bounds L_i and H_i, respectively. Thus a vehicle is inside the intersection if $p_i(t) \in [L_i, H_i]$, and a collision between two vehicles i and j has occurred at time t if $[p_i(t)\, p_j(t)] \in [L_i, H_i] \times [L_j, H_j]$.

Furthermore, vehicles can at any time t generate noisy observations

$$\mathbf{y}_i(t) = \mathbf{x}_i(t) + \mathbf{n}_i(t) \tag{2}$$

of their state and send these to the IM on an uplink (UL) channel. We denote by $\mathbf{y}_i^{\text{tot}}(t)$ the vector of noisy observations regarding vehicle i up to, and including, time t. The IM executes a tracking filter to determine the distribution $p\left(\mathbf{x}_i(t)|\mathbf{y}_i^{\text{tot}}(t)\right)$ of each vehicle's state. Using this, the IM then computes and broadcasts control signals on a downlink (DL) channel.

B. Traffic Controller

The traffic controller operates in receding horizon (with finite time horizon KT_s) fashion, and at discrete times kT_s solves an optimization problem of the form [1]

$$\begin{aligned} \underset{\text{vehicle controls}}{\text{minimize}} \quad & \text{performance criterion} && \text{(3a)} \\ \text{subject to} \quad & \text{dynamics} && \text{(3b)} \\ & \text{safety constraints} && \text{(3c)} \end{aligned}$$

based on the state estimates from the tracking filter. The output of the traffic controller is thus a piecewise constant control over the horizon $[kT_\text{s}, (k+K)T_\text{s}]$, which can be described by

$$u_{i,l}, \text{ for } l = 0, ..., K-1. \tag{4}$$

Note that l here refers to future time steps with respect to the current time index k. The traffic controller is further assumed to know the intersection geometry and vehicle dynamics. The performance criterion in (3a) could include total consumed energy, fuel consumption or deviation from target speed. The safety constraints ensure that collisions are avoided. The traffic controller is an off-the-shelf controller and is not aware of any uncertainties in the system state (i.e., it only knows the expected state) or how the underlying communication works.

Remark 1. Robust control formulations, which directly takes into account uncertainties, could also be considered. Such formulation would need little or no communication, but often lead to an overly conservative behavior (see e.g., [24]), as the control actions are based on predictions with increasing uncertainties over time. Such robust formulations are not considered in this work.

C. Resource Allocator

We introduce the collision possibility indicator (CPI), $C_{i,j}(k,l) \in \{0,1\}$, to indicate, as predicted by the IM at time k, whether the states of vehicle i and j, l time steps in the future may be such that the risk of collision can no longer be excluded. For instance, when $C_{1,2}(k,5) = 1$, then the IM predicts that 5 time steps in the future, vehicles 1 and 2 may be in a configuration that leads them to a future collision. The computation of $C_{i,j}(k,l)$ is based on information available to the resource allocator at time step k, such as uncertainties in the vehicle states as well as the control signals from the traffic controller, and will be described in Section III.

The goal of the resource allocator is to schedule communication between vehicles and IM so as to minimize communication resources, while avoiding possible collisions. Formally this can be expressed as

$$\begin{aligned} \underset{s_{i,l}\ \forall i,l}{\text{minimize}} \quad & \sum_{i=1}^{N} \sum_{l=1}^{K-1} s_{i,l} && \text{(5a)} \\ \text{s.t.} \quad & s_{i,l} \in \{0,1\}, && \forall i,l \quad \text{(5b)} \\ & \sum_{\tilde{l} \le l} s_{i,\tilde{l}} \ge C_{i,j}(k, l+1), && \forall i, j \neq i, l \quad \text{(5c)} \\ & s_{i,l} \le \max_j \{C_{i,j}(k,l+1)\}, && \forall i, j \neq i, l \quad \text{(5d)} \end{aligned}$$

where $s_{i,l}$ indicates whether a certain vehicle i is assigned communication resources at time $(k+l)T_\text{s}$. While the objective is to minimize the amount of time slots used for communication, the constraint (5c) ensures that vehicle i (resp. j) has communicated at least once with the IM before the risk of a collision can no longer be excluded, i.e., before $C_{i,j}(k,l+1) = 1$. Furthermore, the constraint in (5d) makes sure that no communication resources are allocated before it is absolutely necessary.

In other words, we aim to minimize uplink communication resources, while avoiding future collisions, and our goal will be to design a resource allocator that makes sure that future collisions can be avoided while accounting for state uncertainty (from (1)–(2)).

While (5) corresponds to a relatively simple resource allocation problem, it can be generalized to account for bandwidth limitations, deadlines, and randomness of the channel. Such generalizations are beyond the scope of the current work.

III. Characterization of Possible Collisions

In this section we characterize the CPI, i.e., $C_{i,j}(k,l)$, which plays a central role in the optimization problem in Section II-C. We associate with vehicle i the set $\mathcal{S}_i(k) = \mathcal{P}_i(k) \times \mathcal{V}_i(k)$, representing the support of $p\left(\mathbf{x}_i(kT_\text{s})|\mathbf{y}_i^{\text{tot}}(kT_\text{s})\right)$, assuming bounded uncertainties. However, distributions with infinite support could also be considered, but require approximation of the uncertainty to evaluate $C_{i,j}(k,l)$. This could be done by considering expanded uncertainties with a reasonable coverage factor (e.g., 3σ-regions) [25]. Given the control command and the vehicle dynamics (1), we can then describe the set $\mathcal{S}_i(k,l)$, representing the possible values of the vehicle state at time $(k+l)T_\text{s}$ in open loop (i.e., in the absence of further received information), derived from the predictive distribution $p\left(\mathbf{x}_i((k+l)T_\text{s})|\mathbf{y}_i^{\text{tot}}(kT_\text{s})\right)$. Hence, it suffices to determine whether the sets $\mathcal{S}_i(k,l)$ and $\mathcal{S}_j(k,l)$ contain states, which inevitably lead to a collision. As we determine the CPI for any future time $(k+l)T_\text{s}$, we will drop the arguments k and l.

A. Collision in the Absence of Uncertainties

When there are no uncertainties, the sets $\mathcal{S}_i$ and $\mathcal{S}_j$ collapse to points. Thus, we can consider two vehicles with known

combined state $\tilde{\mathbf{x}} = [p_i \ v_i \ p_j \ v_j]^{\mathrm{T}}$ and determine whether this state inevitably leads to collision. To this end, we introduce $\mathbf{x}(t, \tilde{\mathbf{x}}, \mathbf{u})$, denoting the future state at time t, from state $\tilde{\mathbf{x}}$, applying control input sequence denoted by $\mathbf{u}$. Note that $\mathbf{x}(t, \tilde{\mathbf{x}}, \mathbf{u})$ can be seen as the continuous flow of the system [26], [27]. A collision corresponds to the vehicles being in the so-called bad set $\mathcal{B}$, defined as

$$\mathcal{B} = \{\mathbf{x} \mid [p_i \ p_j] \in [L_i, H_i] \times [L_j, H_j]\}. \tag{6}$$

For a collision to occur, the vehicles must be in a state $\tilde{\mathbf{x}}$ at an earlier time, for which a collision is unavoidable. These states are characterized by the so-called capture set [26], [27]

$$\mathcal{C} = \{\tilde{\mathbf{x}} \mid \forall \mathbf{u}, \exists t \in \mathbb{R}^{+} \text{s.t.} \, \mathbf{x}(t, \tilde{\mathbf{x}}, \mathbf{u}) \in \mathcal{B}\}, \tag{7}$$

i.e., the set of states for which, no matter what control input is applied, the vehicles will inevitably end up in a collision. For computational reasons, slices of the capture set will be considered (and also visualized) in position space:

$$\mathcal{C}_{[v_i \ v_j]} = \{[p_i \ p_j] \mid \forall \mathbf{u}, \exists t \in \mathbb{R}^{+}, \text{s.t.} \, \mathbf{x}(t, [p_i \ v_i \ p_j \ v_j]^{\mathrm{T}}, \mathbf{u}) \in \mathcal{B}\}, \tag{8}$$

i.e., the set of positions that for fixed initial velocities v_i and v_j inevitably will lead to a collision no matter what control input is applied. We can thus in the absence of uncertainties express the CPI as

$$C_{i,j} = \begin{cases} 1 & [p_i \ p_j] \in \mathcal{C}_{[v_i \ v_j]} \\ 0 & \text{else} \end{cases}. \tag{9}$$

Note that in the absence of uncertainties $C_{i,j} = 1$ implies that a collision is unavoidable. An example of $\mathcal{C}_{[v_i \ v_j]}$ is shown in Figure 2. We note that the capture set slice shrinks the further the positions of the vehicles are away from the intersection. The size of the capture set slices depends on the allowable control signals (a larger control interval leads to a smaller capture set slice as collisions are easier to avoid) as well as the velocities (with increased velocities of both vehicles, the capture set slice grows, while with increased velocity of one vehicle, the capture set slice will move upward or downward).

B. Collision in the Presence of Uncertainties

In the presence of uncertainties, the sets $\mathcal{S}_i$ and $\mathcal{S}_j$ describe the possible vehicle states. We further decompose these sets into intervals, i.e., $p_i \in \mathcal{P}_i = [p_i^{\mathrm{L}}, p_i^{\mathrm{H}}]$, $p_j \in \mathcal{P}_j = [p_j^{\mathrm{L}}, p_j^{\mathrm{H}}]$, $v_i \in \mathcal{V}_i = [v_i^{\mathrm{L}}, v_i^{\mathrm{H}}]$, $v_j \in \mathcal{V}_j = [v_j^{\mathrm{L}}, v_j^{\mathrm{H}}]$. This allows us to consider uncertainty in the position and velocity separately, and to form the set

$$\mathcal{C}_{\mathcal{V}_i, \mathcal{V}_j} = \bigcup_{v_i \in \mathcal{V}_i, v_j \in \mathcal{V}_j} \mathcal{C}_{[v_i \ v_j]}, \tag{10}$$

which can be interpreted as the set of positions for which there exists a velocity pair $[v_i \ v_j] \in \mathcal{V}_i \times \mathcal{V}_j$ that will inevitably lead to a collision. In other words given the uncertainties in the velocity there is a possibility that a collision may occur

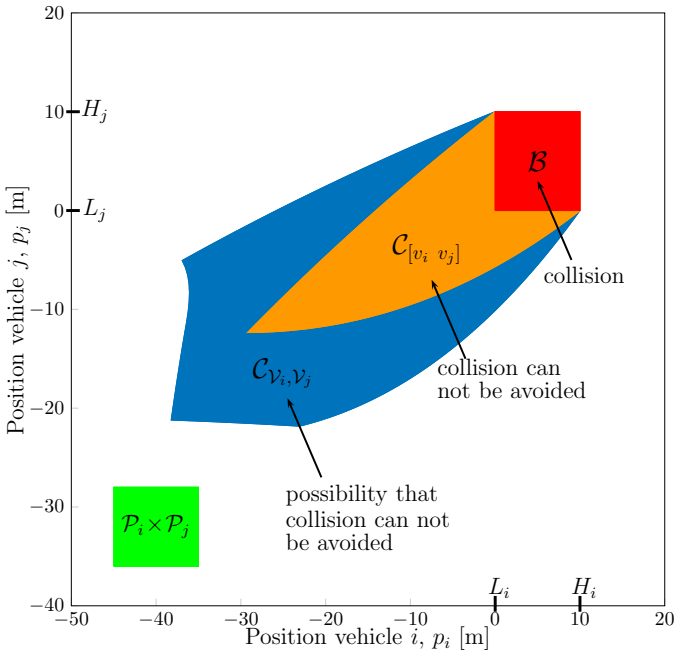


Figure 2. Illustration of the bad set $\mathcal{B}$ in position space, along with an example of $\mathcal{C}_{[v_i \ v_j]}$, $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$, and $\mathcal{P}_i \times \mathcal{P}_j$. The slice $\mathcal{C}_{[v_i \ v_j]}$ is shown for $v_i = 60$ km/h and $v_j = 40$ km/h, while $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$, and $\mathcal{P}_i \times \mathcal{P}_j$ are shown for $\mathcal{V}_i = [56, 64]$ km/h, $\mathcal{V}_j = [36, 44]$ km/h, $\mathcal{P}_i = [-45, -35]$ m and $\mathcal{P}_j = [-36, -28]$. Note that these values are just chosen for illustration purposes and are not the ones used in the simulations in Section V.

when the positions of the vehicles are in the set $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$. Thus, the CPI can be expressed as

$$C_{i,j} = \begin{cases} 1 & \mathcal{P}_i \times \mathcal{P}_j \cap \mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j} \neq \emptyset \\ 0 & \text{else} \end{cases}. \tag{11}$$

An example of $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$ and $\mathcal{P}_i \times \mathcal{P}_j$ is shown in Figure 2. As can be seen, the two sets (visualized in blue and green) do not intersect. Thus $C_{i,j} = 0$, which implies that for this particular example there is at the moment no risk for a future collision. However note that in the presence of uncertainties $C_{i,j} = 1$ (i.e., a non empty intersection) does no longer imply that a collision is unavoidable, only that the possibility of a future collision can not be excluded.

C. General Procedure for Computation of Capture Set Slices

In this section, we will show how to compute $\mathcal{C}_{[v_i \ v_j]}$ and $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$. In particular we will focus on how to characterize the boundaries of the two sets.

1) Computation of $\mathcal{C}_{[v_i \ v_j]}$: It has been shown that for monotone two-vehicle systems, considered here, the system state is steered to $\mathcal{B}$ for all input choices if and only if it is taken to $\mathcal{B}$ both when vehicle i applies maximum brake and vehicle j applies maximum acceleration, and when vehicle j applies maximum brake and vehicle i applies maximum acceleration [26], [27]. Thus, by considering the two restricted capture set slices $\mathcal{C}^{\mathbf{u}_1}_{[v_i \ v_j]}$ and $\mathcal{C}^{\mathbf{u}_2}_{[v_i \ v_j]}$, defined as (8) but fixing the control inputs to $\mathbf{u}_1 = [u_{i,\min} \ u_{j,\max}]^{\mathrm{T}}$ and $\mathbf{u}_2 = [u_{i,\max} \ u_{j,\min}]^{\mathrm{T}}$, we can compute the capture set slice as

$$\mathcal{C}_{[v_i \ v_j]} = \mathcal{C}^{\mathbf{u}_1}_{[v_i \ v_j]} \cap \mathcal{C}^{\mathbf{u}_2}_{[v_i \ v_j]}. \tag{12}$$

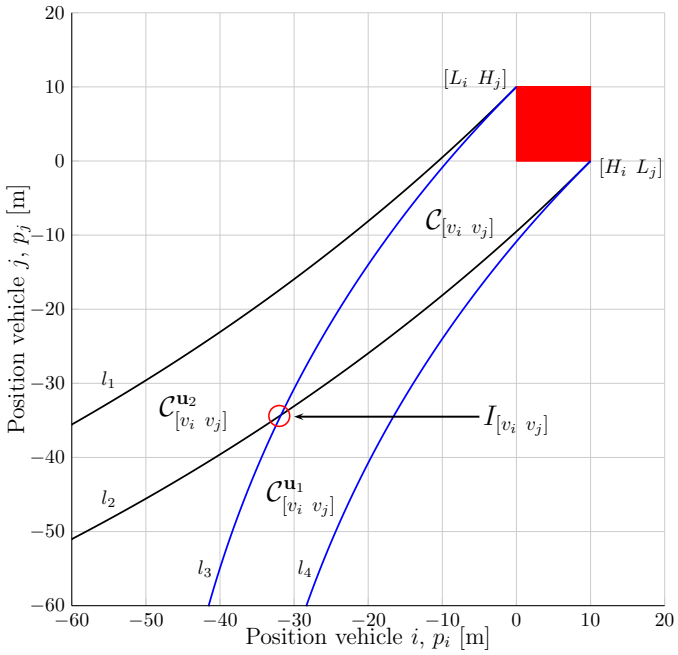


Figure 3. Illustration of the bad set $\mathcal{B}$ in position space, along with the four curves l_1 to l_4, of which l_2 and l_3 characterizes the set $\mathcal{C}_{[v_i\ v_j]}$. The curves are computed for double integrator dynamics, $v_i = 60$ km/h and $v_j = 40$ km/h, $\mathbf{u}_1 = [-1\ \ 1]$ m/s^2, and $\mathbf{u}_2 = [1\ \ -5]$ m/s^2.

Each of these sets can be characterized by two curves, starting from $[L_i\ \ H_j]$ and from $[H_i\ \ L_j]$ back-propagating with the system dynamics and the constant extremal control inputs. An illustration of the sets along with the four curves is shown in Figure 3. As the intersection of the two sets defines $\mathcal{C}_{[v_i\ v_j]}$, it is sufficient to compute the curves l_2 and l_3 and find their intersection point $I_{[v_i\ v_j]}$ in order to characterize its boundary. More specifically, this can be done by either:

- Modifying Algorithm 1 in [26], such that the upper left and lower right corner points are stored when back propagating the bad set. By doing this, and making sure that the algorithm terminates when the corner points (i.e. the curves) get sufficiently close to each other, we obtain both the two curves l_2 and l_3 as well as their intersection point $I_{[v_i\ v_j]}$.
- Using the analytic characterization of the two curves l_2 and l_3 that we provide in Appendix A, in combination with fixed point iteration to find the intersection point $I_{[v_i\ v_j]}$.

The latter approach provides a more efficient way to characterize the boundary of the capture set slice. However, it is in comparison to the first approach less general, as the analytic descriptions of the curves l_2 and l_3 provided in Appendix A are for the specific case of double integrator dynamics.

2) Computation of $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$: Since $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$ is not necessarily convex and its computation involves infinitely many sets, it is in general hard to compute. However, by discretizing the velocity uncertainty intervals and sweeping over the different speed combinations we can visualize $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$, see Figure 4. We then note the following about $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$:

- The set itself, which is visualized in blue, is the union of infinitely many capture set slices in position space. Five of these are visualized using orange dashed lines.
- Its boundary is characterized by the curves $\tilde{l}_2$ and $\tilde{l}_3$, as well as the two curves obtained by tracing the boundary of the blue set between the three intersection points $I_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{L}}]}$, $I_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{H}}]}$, and $I_{[v_i^{\mathrm{L}}\ v_j^{\mathrm{H}}]}$. For ease of notation, we denote the latter two curves C_1 and C_2, where C_1 corresponds to the curve between $I_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{L}}]}$ and $I_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{H}}]}$, and C_2 corresponds to the curve between $I_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{H}}]}$, and $I_{[v_i^{\mathrm{L}}\ v_j^{\mathrm{H}}]}$.
- The curve $\tilde{l}_2$ corresponds to the part of the previously defined l_2 curve that characterizes the capture set slice $\mathcal{C}_{[v_i^{\mathrm{L}}\ v_j^{\mathrm{H}}]}$, i.e., the part between $[H_i\ \ L_j]$ and $I_{[v_i^{\mathrm{L}}\ v_j^{\mathrm{H}}]}$. Similarly, $\tilde{l}_3$ is the part of the l_3 curve that characterizes the capture set slice $\mathcal{C}_{[v_i^{\mathrm{H}}\ v_j^{\mathrm{L}}]}$. Discretized versions, i.e., point representations, of these curves can thus be computed using the method described in Section III-C1. The discretized versions of the curves are denoted $\tilde{l}_2^{\mathrm{d}}$ and $\tilde{l}_3^{\mathrm{d}}$.
- Discretized versions of the curves C_1 and C_2, denoted C_1^{d} and C_2^{d}, can be obtained by discretizing the velocity uncertainty intervals, and then compute intersection points while keeping one of the velocities fixed at its maximum and sweeping the other over the discretized interval. An example of how the discretized curves could look is shown in the green approximation of the boundary.
- Based on the discretized curves for the boundary we can then efficiently approximate $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$ as

$$\mathcal{C}^{\mathrm{approx}}_{\mathcal{V}_i,\mathcal{V}_j} = \mathrm{Conv}\left(\tilde{l}_2^{\mathrm{d}} \cup \tilde{l}_3^{\mathrm{d}} \cup C_1^{\mathrm{d}} \cup C_2^{\mathrm{d}}\right). \tag{13}$$

Remark 2. The approach presented above generally provides a tighter approximation of $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$ compared to if state uncertainties are considered as described in [26].

IV. Collision-Aware Allocation of Communication Resources

Until here we have described the basic properties of the system and characterized the CPI, i.e., the possibility of collision between two vehicles. Based on this, we are now ready to put everything together and explain the operation of the IM, and detail a procedure for collision-aware resource allocation (CARA) for the N vehicle scenario outlined in Section II. Without loss of generality we will focus on an arbitrary time instance kT_{s} to describe the operation of the resource allocator and the IM.

A. Resource Allocator

To start with, we assume that at time kT_{s} the resource allocator has access to the output from the traffic controller, i.e., control signals $[u_{i,0}\ \cdots\ u_{i,K-1}]$ for all vehicles $i = 1, ..., N$. Furthermore, the resource allocator has access to the uncertainty in the vehicle states, in the form of sets $\mathcal{S}_i(k)$, which describe the possible states of all vehicles at the initial time kT_{s}. Recall that while the control signals are purely based

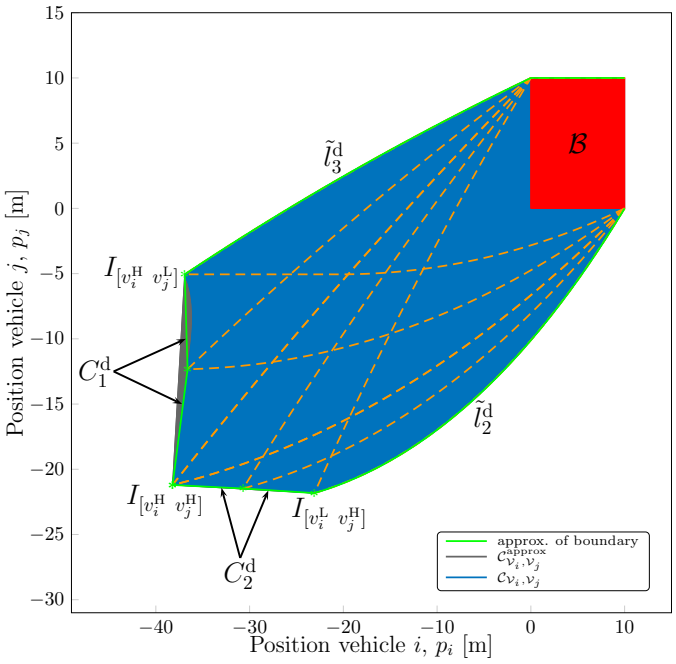


Figure 4. Illustration of $\mathcal{C}_{\mathcal{V}_i,\mathcal{V}_j}$ and its approximation $\mathcal{C}^{\mathrm{approx}}_{\mathcal{V}_i,\mathcal{V}_j}$, which can be efficiently computed. The green curve is an approximation of the boundary based on the discretized curves $\tilde{l}_2^{\mathrm{d}}$, $\tilde{l}_3^{\mathrm{d}}$, C_1^{d} and C_2^{d}.

on expected states, the sets $\mathcal{S}_i(k)$ are based on the distribution $p\left(\mathbf{x}_i(kT_{\mathrm{s}})|\mathbf{y}_i^{\mathrm{tot}}(kT_{\mathrm{s}})\right)$. The resource allocator takes the control signals and the sets $\mathcal{S}_i(k)$ as input and executes the following procedure. First, it uses the control signals along with the initial state uncertainty to compute the predictive distribution, and to form the sets $\mathcal{S}_i(k,l)$ for $i = 1,...,N$ and $l = 1,...,K$, i.e., it computes possible future states for all time instances along the controller prediction horizon. Based on these sets and the vehicle dynamics it then computes $C_{i,j}(k,l)$ for $i = 1...N$, $j = i+1...N$ and $l = 1,...,K$ using the approximation introduced in (13). In other words it tabulates the CPI for all vehicle pairs and time instances along the prediction horizon. After doing this the resource allocator solves the integer program (IP) in (5)[1]. The output from the IP, which is $s_{i,l}^*\ \forall i,l$, contains information about the latest possible time that each vehicle need to send updated information in order to make sure that future collisions can be avoided. The steps of the resource allocator are summarized in Algorithm 1.

B. Receding horizon IM

Based on the output form the traffic controller and the resource allocator the IM then, at time kT_{s}, broadcasts information about when each vehicle is expected to communicate the next time, along with control signals for each vehicle up until then. However, as the possibility of collision is evaluated pairwise between all vehicles and depends on the uncertainty in the state of the involved vehicles, the assigned communication slots might change when the IM receives updated information from the vehicles which are scheduled to transmit earliest. For practical reasons, the proposed IM thus operates in a receding horizon fashion, where both the resource allocation and the control signals are revised in the first time slot where updated information is received from any of the vehicles. This means that in practice all vehicles except the ones assigned to communicate first will receive updated instructions before it is their time to communicate according to the time slot assignment performed at time kT_{s}. The operation of the IM is summarized in Algorithm 2.

[1]Although IPs are generally NP-complete, (5) can be solved efficiently in a time $\mathcal{O}(N^2K)$, as will be illustrated in Section IV-C

Algorithm 1 Resource Allocator

Input: current state sets $\mathcal{S}_i(k)$, control vector $[u_{i,0}\ \cdots\ u_{i,K-1}]$, and system parameters $\mathbf{A}_i$,$\mathbf{b}_i$,L_i,H_i and $[u_{i,\max}, u_{i,\min}]$ for $i = 1,...,N$.
Output: $s_{i,l}$ for $i = 1,...,N$ and $l = 1,...,K-1$
1: Compute $\mathcal{S}_i(k,l)$ for $l = 1,...,K$ and $i = 1,...,N$
2: Tabulate $C_{i,j}(k,l)$ for $l = 1,...,K, i = 1,...,N$ and $j = i+1,...,N$
3: Solve IP in (5)
4: **return** $s_{i,l}^*\ \forall i,l$

Algorithm 2 IM

1: **for** each time step k **do**
2: **if** new measurements are received **then**
3: Traffic Controller computes $u_{i,l}$ for $l = 0,...,K-1, \forall i$
4: $[s_{i,l}^*]_{\forall i,l}$=Resource Allocator($[\mathcal{S}_i(k)\ u_{i,l}]_{\forall i,l}$)
5: Send $[s_{i,l}^*\ u_{i,l}]_{\forall l}$ to vehicle i, $\forall i$
6: **end if**
7: **end for**

C. Example: 4 Vehicle Scenario

To get some intuition on how the collision-aware allocation of communication resources works, we will now consider a scenario with $N = 4$ vehicles, which at time kT_{s} are located as shown in Figure 1, i.e., three vehicles with approximately the same distance to the intersection and a fourth vehicle further away from the intersection. For simplicity, we assume that all four vehicles previously have communicated with the IM and that the current uncertainties about the vehicle states are approximately the same. As previously described, the traffic controller first computes control signals for the whole prediction horizon. This information along with the uncertainty in the vehicle states are then passed along to the resource allocator, which propagates the uncertainty along the prediction horizon and evaluates the CPI for all time instances along the prediction horizon (Algorithm 1, step 2). An illustration of how this could look, given the assumption that the initial uncertainties are approximately the same, and that the controller due to initial locations of vehicle 1, 2 and 4 has scheduled them to cross in rapid succession, is shown in Table I. From this table we see, that the CPI is zero for all vehicle pairs in the first three time steps along the prediction horizon. However, at time step $l = 4$ in the future,

Table I
PAIRWISE POSSIBILITIES OF COLLISION ALONG THE PREDICTION HORIZON COMPUTED AT TIME STEP k.

	1	2	3	4	5	6	7	8	...	K
$C_{1,2}(k,l)$	0	0	0	1	1	1	1	1		1
$C_{1,3}(k,l)$	0	0	0	0	0	0	1	1		1
$C_{1,4}(k,l)$	0	0	0	1	1	1	1	1		1
$C_{2,3}(k,l)$	0	0	0	0	0	0	1	1		1
$C_{2,4}(k,l)$	0	0	0	1	1	1	1	1		1
$C_{3,4}(k,l)$	0	0	0	0	0	0	1	1		1

Table II
RESULTING TIME SLOT ALLOCATION AT TIME STEP k.

	1	2	3	4	5	6	7	8	...	K
$s^*_{1,l}$	0	0	1	0	0	0	0	0		0
$s^*_{2,l}$	0	0	1	0	0	0	0	0		0
$s^*_{3,l}$	0	0	0	0	0	1	0	0		0
$s^*_{4,l}$	0	0	1	0	0	0	0	0		0

the uncertainties, represented by sets of the form $\mathcal{S}_i(k,4)$, of vehicles 1, 2 and 4, have increased to such an extent that a collision can no longer be ruled out. At time step $l = 7$, additionally, the uncertainties are so high that potential collisions can occur between all vehicle pairs.

Correspondingly, Table II shows the resulting slot assignment obtained by solving the IP in (5). To relate this table to Table I, we first observe that as long as all the pairwise CPIs are zero, none of the vehicles are assigned communication resources in the previous time slots. Then, we note that in the first time slot that the CPI between any two vehicles become non-zero, both of these vehicles are assigned communication resources in the time slot before this, given that they haven't already been assigned to communicate in an earlier time slot. Thus, as the CPI between vehicle 1, 2, and 4 turn one in slot $l = 4$, all three vehicles are assigned to communicate in slot $l = 3$. Similarly, vehicle 3 is finally assigned to communicate in slot $l = 6$. This procedure can be generalized to arbitrary K and N, leading to a complexity of $\mathcal{O}(N^2K)$. The assigned communication slots along with the computed control signals are then sent to the respective vehicles. However, note that due to the receding horizon operation of the IM, vehicle 3 will not necessarily communicate in time slot $l = 6$, as both the control signals and the slot assignment will be revised at time $(k+3)T_\mathrm{s}$ when the IM receives updated information from vehicle 1, 2, and 4.

V. NUMERICAL RESULTS

To more thoroughly demonstrate the concept of the collision-aware resource allocation strategy, and to be able to evaluate its benefits and drawbacks, we will study and compare the impact of the following communication strategies:

- Baseline: The vehicles communicate updated information every T_s. This is the situation that the controller was designed for.
- Low rate: The vehicles communicate updated information every $10T_\mathrm{s}$.
- CARA: The vehicles communicate updated information according to the collision-aware resource allocation strategy presented in Section IV.
- M-CARA: This is a modified version of the CARA protocol where a deadline time T_d is the maximum time that the vehicles wait to communicate updated information to the IM. However, if there is a risk for a future collision before the deadline has expired, the vehicles are assigned to communicate earlier.

A. Simulation Setup

We consider an intersection scenario with $N = 2$ vehicles. The simulations are discretized on a uniform time grid with sample time $T_\mathrm{s} = 0.1\,\mathrm{s}$ and discrete time index k. The discrete states $\mathbf{x}_{i,k} = [p_{i,k}\ v_{i,k}]^\mathrm{T}$of the two vehicles are assumed to evolve according to double integrator dynamics, i.e.,

$$\mathbf{x}_{i,k+1} = \underbrace{\begin{bmatrix}1 & T_\mathrm{s}\\ 0 & 1\end{bmatrix}}_{\mathbf{A}_i^\mathrm{d}}\mathbf{x}_{i,k} + \underbrace{\begin{bmatrix}T_\mathrm{s}^2/2\\ T_\mathrm{s}\end{bmatrix}}_{\mathbf{b}_i^\mathrm{d}}u_{i,k} + \mathbf{w}_{i,k}, \qquad (14)$$

where $\mathbf{A}_i^\mathrm{d}$ and $\mathbf{b}_i^\mathrm{d}$ are the discrete counter parts of $\mathbf{A}_i$ and $\mathbf{b}_i$ in (1), the control signal $u_{i,k}$ correspond to the demanded acceleration, and $\mathbf{w}_{i,k}$ is discretized process noise. For the traffic controller, we use the receding horizon controller (RHC) presented in [20, Problem 11], with stage cost

$$\tau(k) = \sum_{i=1}^{N}(v_{i,k} - v_i^\mathrm{ref})^2 Q + u_{i,k}^2 R,$$

which penalizes deviations from a desired reference speed v_i^ref as well as control actions. We ensure that vehicle 2 passes closely after vehicle 1 by adding a constraint

$$t_2^\mathrm{in} \in [t_1^\mathrm{out} + \varepsilon, t_1^\mathrm{out} + 2\varepsilon], \qquad (15)$$

where t_1^out is the time vehicle 1 clears the intersection, t_2^in is the time vehicle 2 enters the intersection, and ε is a safety padding, which is set to achieve aggressive yet safe behavior under the baseline communication strategy. To avoid infeasibility, (15) is implemented as a soft constraint [28]. The horizon length is set to $K = 100$ steps. Furthermore, we let $L_i = 0$ m and $H_i = 10$ m, and the initial conditions of the vehicles are set such that the two vehicles will collide if no control action is taken. More precisely, both vehicles start 150 m away from the intersection with a speed of 70 km/h. The desired reference speed $v_i^\mathrm{ref} = 70$ km/h, i.e., the desired behavior is that the vehicles deviate as little as possible from their initial speed. Also, we assume that independently of which communication strategy we apply, the vehicles always send observations of their initial state to the IM at $k = 0$. The discrete process noise (which describes the mismatch between

Table III
SIMULATION PARAMETERS

Parameters	Values
Number of vehicles	$N = 2$
Sample time (s)	$T_\mathrm{s} = 0.1$
Horizon length	$K = 100$
Safety padding (s)	$\varepsilon = 0.02$
Lower bound of intersection (m)	$L_i = 0$
Upper bound of intersection (m)	$H_i = 10$
Penalty on deviation from v_i^{ref} [20]	$Q = 1$
Penalty on $u_{i,k}$ [20]	$R = 200$
Actuator constraints (m/s^2)	$u_{i,k} \in [-2\ 2]$
Speed constraints (km/h)	$v_{i,k} \geq 0$
Initial position along trajectory (m)	$p_{i,0} = -150$
Initial speed (km/h)	$v_{i,0} = 70$
Reference speed (km/h)	$v_i^{\mathrm{ref}} = 70$
Process noise on speed (km/h)	$w_\mathrm{s} = 0.36$
Observation noise on position (m)	$n_\mathrm{p} = 0.25$
Observation noise on speed (km/h)	$n_\mathrm{s} = 0.5$

the controller model and the true dynamics) and observation noise are drawn from uniform distributions:

$$\mathbf{w}_{i,k} \sim \mathcal{U}\left(\begin{bmatrix} 0 \\ -w_\mathrm{s} \end{bmatrix}, \begin{bmatrix} 0 \\ w_\mathrm{s} \end{bmatrix}\right), \tag{16}$$

and

$$\mathbf{n}_{i,k} \sim \mathcal{U}\left(\begin{bmatrix} -n_\mathrm{p} \\ -n_\mathrm{s} \end{bmatrix}, \begin{bmatrix} n_\mathrm{p} \\ n_\mathrm{s} \end{bmatrix}\right), \tag{17}$$

where values on w_s, n_p and n_s are found in Table III.

We observe that, while the considered simulation setup is limited to two vehicles for the sake of presentation, our approach can be extended to any number of vehicles. Clearly, the sets $C_{i,j}$ must be calculated for all possible colliding vehicles and Problem (5) grows in size, hence, in complexity.

B. Performance metrics

In the following section we present results from a set of Monte Carlo simulations. However, before presenting these results, we briefly introduce the performance metrics that we will use to evaluate and compare the different communication strategies:

- **Probability of communication** relates to a specific time slot k and is defined as the fraction of realizations in which the vehicles communicate in this time slot.
- **Average control cost** relates to a specific time slot k, and is defined as the stage cost $\tau(k)$ evaluated on the true vehicle states averaged over the number of Monte Carlo realizations.
- **Total average control cost** is defined as the integral of the average control cost, i.e, the total accumulated average control cost over the length of the simulation.
- **Average number of communication instances** is defined as the average number of times the vehicles communicate until they have passed the intersection.
- **Number of collisions** is defined as the total number of recorded collisions in a set of Monte Carlo Simulations.

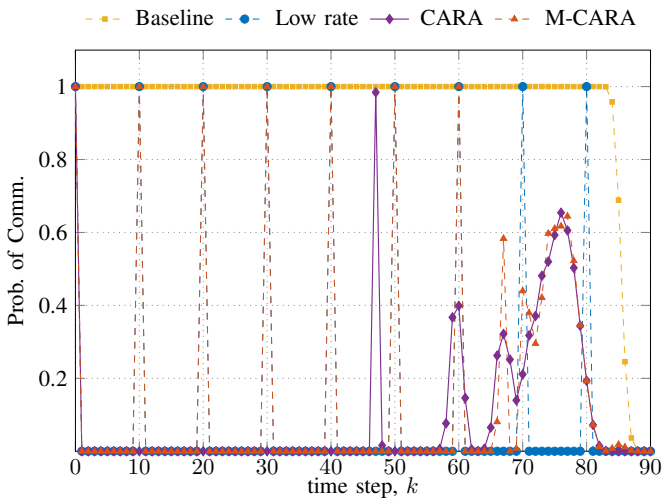
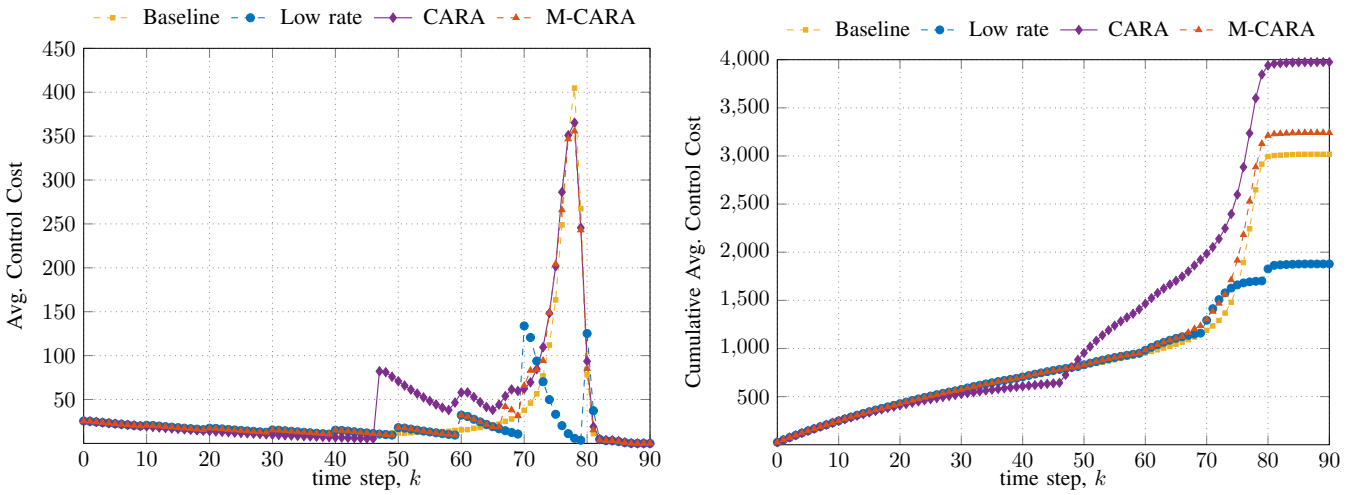


Figure 5. Probability of communication as a function of time for the four different strategies. Each simulated point is based on 10000 realizations.

C. Results and Discussion

The performance of the different strategies are evaluated and compared by running Monte Carlo simulations with 10 000 realizations for each of the four communication strategies.

1) Communication Performance: To understand the communication behavior (i.e., when the vehicles communicate), we visualize the probability of communication as a function of time, see Figure 5. We observe that, as expected, the low rate strategy communicates every $10T_\mathrm{s}$ and the baseline strategy communicates every T_s, except towards the end, as the time for both vehicles to clear the intersection can vary depending on the noise realizations. Under the CARA and M-CARA strategies, vehicles communicate much less compared to the baseline strategy, and in contrast to the low rate strategy, the CARA strategy initially requires less communication and later allocates more communication resources as the vehicles gets closer to the intersection and thus are in a more critical condition. In particular, we observe a communication peak around time step $k = 47$, corresponding to the first time (after the initial communication at $k = 0$, which is the starting point for the propagation of the uncertainties and the computation of the capture set) where a collision can occur. The next time updated information is sent to the IM is around time step $k = 60$, where we note a wider peak. The reason for this is the effect the process noise has on the trajectory of the two vehicles. More specifically, if the results of the process noise is a trajectory that is further away from the capture set around time step $k = 47$, the vehicles can safely wait longer to send updated information to the IM. On the other hand, if the trajectory is closer to the capture set, the vehicles need to communicate sooner. After the three first peaks, there is a wide peak before $k = 80$. This peak can be explained by the fact that when the vehicles are very close to the intersection, they need to communicate frequently to maintain sufficiently low uncertainties to guarantee safe passage through the intersection. The M-CARA strategy behaves similar to



(a) (b)

Figure 6. Control performance in terms of average control cost (a) and cumulative average control cost (b) as a function of time for the our different strategies. Each simulated point is based on 10000 realizations.

the CARA strategy, except that the longest gap between communications is limited to $T_{\mathrm{d}} = 10T_{\mathrm{s}}$. Finally, we also note that the probability of communication drops off earlier for the CARA and M-CARA, since in both strategies there is no need for communication as soon as the first vehicle has cleared the intersection area, since there is no longer any risk for a future collision.

2) Control Performance: To evaluate the impact the different strategies have on the control of the two vehicles, Figure 6 shows the average control cost as well as the cumulative average control cost. From Figure 6a, we observe that in general the average control cost is low to start with and then significantly increases as the vehicles get closer to the intersection. This behavior is due to the specific setup, and the fact that both process and observation noise have a larger impact as the vehicles gets closer to the intersection, as the relatively shorter time to compensate for potential deviation requires larger control actions. Furthermore, we see that all strategies perform rather similar in the beginning, but that strategies where less communication resources are assigned lead to a lower average control cost towards the end. This is an effect of how the traffic controller is implemented, as more feedback will result in a t_2^{in} that is close to $t_1^{\mathrm{out}} + \varepsilon$ and thus a higher noise sensitivity. Moreover, we observe that for the CARA strategy we have a larger average control cost when the IM receives and update from the vehicles, which is due to the fact that it is better to be conservative and immediately apply a large control action to correct potential deviations than letting the error accumulate over time. Finally, we also see in Figure 6b, that the total average control cost is highest for the CARA strategy and lowest for the low rate strategy.

3) Collision Occurrence: The above results may seem to indicate that the low rate strategy is favorable from both a communication and control perspective. However, Table IV, which puts the communication and control performance in relation to the number collisions for the different strategies, demonstrates that the low rate strategy does not meet the basic safety requirement. On the other hand, we see that the CARA and M-CARA strategies reduce the amount of required communication to similar levels as the low rate strategy, but maintain the safety, i.e., the number of collision is zero for both CARA and M-CARA. In fact, results not included here have shown that even if the process noise is increased (without adjusting the safety padding) such that the collision rate becomes nonzero, the collision rate is similar for the baseline, CARA and M-CARA strategies, while it is significantly worse for the low rate strategy. Furthermore, from this table we also note that the average number of communications instances required are rather similar for the CARA and M-CARA strategies in relation to the baseline, even though the M-CARA strategy interestingly has a lower total average control cost than CARA. In fact, it can be seen in Figure 6 that in terms of control performance, the M-CARA strategy performs similar to the baseline strategy, both in terms of average control cost and cumulative average control cost.

Table IV
RESULTS

Parameters	Baseline	Low Rate	CARA	M-CARA
Total Avg. Control Cost	3018	1878	3975	3241
Avg. No. of Comm. Inst.	85.9	9.0	8.9	12.9
No. Collisions	0	155	0	0

4) Communication / Control Trade-off: To further investigate the fact that the M-CARA strategy has a lower total average control cost compared to CARA, but still a significant reduction in the required amount of communication, we study the relationship between total average control cost and average number of communication instances required for safe operation by varying the deadline time T_{d}. The result is shown in Figure 7, which indicates that more communication generally leads to a lower total average control cost. However, for the specific scenario under consideration, we observe that

in terms of total average control cost there is an optimal deadline time $T_{\mathrm{d}} = 3T_{\mathrm{s}}$. This means that there is no meaning in communicating more than this. Furthermore, for $T_{\mathrm{d}} > 3T_{\mathrm{s}}$, we observe a sharp increase in total average control cost. This implies that CARA, which was designed to minimize the communication load while avoiding collisions, is not optimal in terms of total average control cost. On the other hand, M-CARA can be applied to trade off control performance and communication load, while still maintaining safety.

VI. Conclusions

We have presented a novel strategy for minimizing the use of communication resources for intersection management of automated and connected vehicles. The communication strategy relies on a new concept referred to as collision possibility indicator (CPI), which characterizes risk of future collisions for pairs of vehicles. The CPI accounts for state uncertainty as well as the dynamics of the vehicles. By evaluating the CPIs over a prediction horizon, we establish when it is necessary to communicate in order to rule out the possibility of future collisions. The proposed strategy, which allocates communication resources in a self-triggered (proactive) manner based on the CPIs, is able to maintain safety while significantly reducing the required amount of communication compared to the baseline scenario with fixed transmission intervals. Since the proposed strategy leads to an increase in the control cost compared to the baseline scenario, we also present a modified strategy where control performance can be traded for increased communication load. In other words, we show how the proposed strategy can be adapted to both scenarios where communication resources are scarce, and scenarios where communication resources are abundant and control performance, such as low energy consumption or passenger comfort, can be prioritized.

Possible avenues for future research include: (i) incorporation of bandwidth constraints such that only a limited number of agents can transmit in each time slot; (ii) making the collision-aware resource allocation strategy robust to communication imperfections such as packet drops and delays; (iii) investigate the feasibility for scenarios with multiple vehicles on the same path; (iv) explicitly accounting for control performance in addition to collision possibilities.

Appendix A
Analytic curves

To derive analytic expression for the curves l_2 and l_3 presented in Figure 3, we start by letting $\Phi_i(t, v_i, u)$ denote the position of vehicle i at time t, provided that the vehicle starts at position $p_i = 0$ at time zero with velocity v_i and constant input u_i. Given double integrator dynamics, we can thus describe the evolution of the position when we apply maximum brake (i.e., $u_{i,\min}$) as

$$\Phi_i(t, v_i, u_{i,\min}) = \begin{cases} v_i t + 0.5 u_{i,\min} t^2 & \forall t \leq -v_i/u_{i,\min} \\ -v_i^2/(2u_{i,\min}) & \forall t \geq -v_i/u_{i,\min} \end{cases}. \tag{18}$$

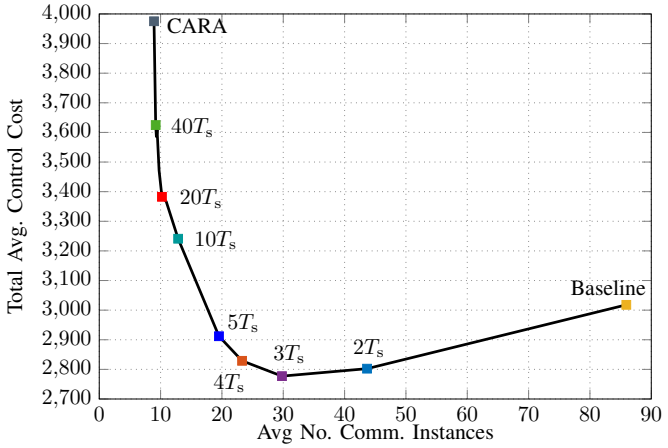


Figure 7. Trade-off between average total cost and average number of communication instances obtained by varying the deadline time T_{d}. Each simulated point is based on 10000 realizations. For none of these results, collisions occurred.

Note that the time $t = -v_i/u_{i,\min}$ corresponds to the time that the vehicle speed reaches zero. Similarly if we apply maximum acceleration the position evolve as

$$\Phi_i(t, v_i, u_{i,\min}) = v_i t + 0.5 u_{i,\max} t^2, \tag{19}$$

where we for simplicity of notation have assumed that there is no upper limit on the vehicle speed. Furthermore, we introduce

$$L_i(t, v_i, u_i) = L_i - \Phi_i(t, v_i, u_i), \tag{20}$$

and

$$H_i(t, v_i, u_i) = H_i - \Phi_i(t, v_i, u_i), \tag{21}$$

which can be interpreted as the backward integration of the upper and lower bounds, given initial velocities v_i and constant control inputs u_i. We are now ready to describe the restricted capture set slices. According to [26, Algorithm 1] they can be computed by propagating the bad set $\mathcal{B}$, which here is the rectangle set characterized by the upper and lower bounds L_i and H_i, backwards with constant extremal control inputs. Hence we get that for vehicle i and j,

$$\mathcal{C}^{\mathbf{u}_1}_{[v_i\, v_j]} = \bigcup_t \left([L_i(t, v_i, u_{i,\min}), H_i(t, v_i, u_{i,\min})] \times [L_j(t, v_j, u_{j,\max}), H_j(t, v_j, u_{j,\max})] \right), \tag{22}$$

and

$$\mathcal{C}^{\mathbf{u}_2}_{[v_i\, v_j]} = \bigcup_t \left([L_i(t, v_i, u_{i,\max}), H_i(t, v_i, u_{i,\max})] \times [L_j(t, v_j, u_{j,\min}), H_j(t, v_j, u_{j,\min})] \right). \tag{23}$$

From this we directly see that the curves l_2 and l_3 can be characterized as

$$l_2 = (H_i(t, v_i, u_{i,\max}), L_j(t, v_j, u_{j,\min})), \forall t, \tag{24}$$

and

$$l_3 = (L_i(t, v_i, u_{i,\min}), H_j(t, v_j, u_{j,\max})), \forall t. \tag{25}$$

Note that the above curves are parameterized by time. By using (18)–(21) and re-organizing the terms we can re-parameterize the curve l_2 such that we get p_j (i.e., the position of vehicle j) as a function of p_i (i.e., the position of vehicle i):

$$p_j = \begin{cases} -v_j\tilde{p}_i - 0.5u_{j,\min}\tilde{p}_i^2 + L_j, & E \leq p_i \leq H_i \\ v_j^2/(2u_{j,\min}) + L_j & p_i \leq E \end{cases}, \tag{26}$$

where $E = H_i + v_i v_j/u_{j,\min} - 0.5u_{i,\max}v_j^2/u_{j,\min}^2$ and $\tilde{p}_i = -v_i\sqrt{v_i^2 - 2u_{i,\max}(p_i - H_i)}/u_{i,\max}$. Similarly we can express the curve l_3 as

$$p_i = \begin{cases} -v_i\tilde{p}_j - 0.5u_{i,\min}\tilde{p}_j^2 + L_i, & F \leq p_j \leq H_j \\ v_i^2/(2u_{i,\min}) + L_i & p_j \leq F \end{cases}, \tag{27}$$

where $F = H_j + v_j v_i/u_{i,\min} - 0.5u_{i,\max}v_i^2/u_{i,\min}^2$ and $\tilde{p}_j = -v_i\sqrt{v_j^2 - 2u_{j,\max}(p_j - H_j)}/u_{j,\max}$.

REFERENCES

[1] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, Nov. 2016.

[2] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[3] G. R. de Campos, P. Falcone, R. Hult, H. Wymeersch, and J. Sjöberg, "Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, pp. 8–21, Spring 2017.

[4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[5] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[6] H. Wymeersch, G. R. de Campos, P. Falcone, L. Svensson, and E. G. Ström, "Challenges for cooperative its: Improving road safety through the integration of wireless communications, control, and positioning," in *Computing, Networking and Communications (ICNC), 2015 International Conference on*, Feb. 2015, pp. 573–578.

[7] M. H. Mamduhi, M. Kneissl, and S. Hirche, "Decentralized event-triggered medium access control for networked control systems," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec. 2016, pp. 513–519.

[8] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, Dec. 2009, pp. 5832–5838.

[9] A. Molin, C. Ramesh, H. Esen, and K. H. Johansson, "Innovations-based priority assignment for control over can-like networks," in *2015 54th IEEE Conference on Decision and Control (CDC)*, Dec. 2015, pp. 4163–4169.

[10] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Control-aware random access communication," in *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, Apr. 2016.

[11] A. Molin, H. Esen, and K. H. Johansson, "Event-triggered scheduling for infrastructure-supported collaborative vehicle control," *IFAC-PapersOnLine*, vol. 49, no. 22, pp. 31–36, 2016.

[12] J. Wu, Q. S. Jia, K. H. Johansson, and L. Shi, "Event-based sensor data scheduling: Trade-off between communication rate and estimation quality," *IEEE Transactions on Automatic Control*, vol. 58, no. 4, pp. 1041–1046, Apr. 2013.

[13] B. Demirel, V. Gupta, D. E. Quevedo, and M. Johansson, "On the trade-off between communication and control cost in event-triggered dead-beat control," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2973–2980, June 2017.

[14] B. Hu, "Self-triggering in Vehicular Networked Systems with State-dependent Bursty Fading Channels," *ArXiv e-prints*, Aug. 2017. [Online]. Available: http://arxiv.org/abs/1708.02347

[15] T. Charalambous, A. Ozcelikkale, M. Zanon, P. Falcone, and H. Wymeersch, "On the resource allocation problem in wireless networked control system," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2017.

[16] C. Nowzari and J. Cortés, "Team-triggered coordination for real-time control of networked cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 1, pp. 34–47, Jan. 2016.

[17] M. Vilgelm, O. Ayan, S. Zoppi, and W. Kellerer, "Control-aware uplink resource allocation for cyber-physical systems in wireless networks," in *European Wireless 2017; 23th European Wireless Conference*, May 2017, pp. 1–7.

[18] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940 – 5946, 2017, 20th IFAC World Congress.

[19] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, 2018.

[20] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec. 2016, pp. 2567–2573.

[21] 5G PPP Architecture Working Group, "View on 5G architecture (Version 2.0)," *White Paper*, Jul. 2017. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2017/07/5G-PPP-5G-Architecture-White-Paper-2-Summer-2017_For-Public-Consultation.pdf

[22] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, 2015.

[23] I. Batkovic, M. Zanon, M. Ali, and P. Falcone, "Real-time constrained trajectory planning and vehicle control for proactive autonomous driving with road users," submitted to *European Control Conference (ECC) 2019*.

[24] A. Bemporad, "Reducing conservativeness in predictive control of constrained systems with disturbances," in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, vol. 2, Dec. 1998, pp. 1384–1389.

[25] Joint Committee for Guides in Metrology (JCGM/WG 1), "Evaluation of measurement data - guide to the expression of uncertainty in measurement," jCGM 100:2008, GUM 1995 with minor corrections, First Edition 2008, Corrected version 2010, Bureau International des Poids et Mesures, France. [Online]. Available: http://www.bipm.org/utils/common/documents/jcgm/JCGM_100_2008_E.pdf

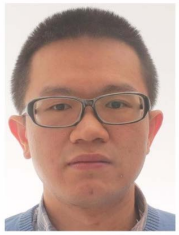
[26] M. R. Hafner and D. D. Vecchio, "Computational tools for the safety control of a class of piecewise continuous systems with imperfect information on a partial order," *SIAM Journal on Control and Optimization*, vol. 49, no. 6, pp. 2463–2493, 2011.

[27] M. R. Hafner, D. Cunningham, L. Caminiti, and D. D. Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1162–1175, Sep. 2013.

[28] J. Maciejowski, *Predictive Control with Constraints*. England.: Prentice Hall, 2002.

Erik Steinmetz received his M.Sc. degree in Electrical Engineering from Chalmers University of Technology, Sweden, in 2009. He is currently a research and development engineer with RISE Research Institutes of Sweden. He is also affiliated with the Department of Electrical Engineering at Chalmers University of Technology, where he is working towards his Ph.D. degree. His research interests include positioning, sensor fusion, communication and controls applied within the fields of intelligent vehicles and cooperative automated driving.

Robert Hult received B.S. degree in Mechanical Engineering in 2011, and the M.Sc. in Systems, Control and Mechatronics in 2013, both from Chalmers University of Technology, Sweden, where he is currently pursuing the Ph.D. degree. His current research interests include distributed and cooperative predictive control, in particular with applications to cooperative vehicles and intelligent transportation systems.



Zhenhua Zou received his Ph.D. degree at the School of Electrical Engineering, KTH The Royal Institute of Technology, Sweden in 2014. He was a postdoc researcher at Chalmers University of technology in 2015. His research interest was about control, optimization and learning algorithms over communication networks. He is now at Ericsson Research.

Ragne Emardson received his M.Sc. in Computer Science and Engineering and Ph.D. in Electrical Engineering from Chalmers University of Technology in 1992 and 1998 respectively. From 1998 to 2000, he was a postdoctoral researcher with the Jet Propulsion Laboratory, California Institute of Technology. He has also held positions with Ericsson Mobile Data Design, Saab Ericsson Space and RISE Research Institutes of Sweden. Currently, he is dean of faculty at University of Borås. His research interests include Global Navigation Satellite Systems, mathematical modeling, and measurement uncertainty evaluation.

Fredrik Brännström received the M.Sc. degree from Luleå University of Technology, Luleå, Sweden, in 1998, and the Ph.D. degree in Communication Theory from the Department of Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden, in 2004. From 2004 to 2006, he was a Post-Doctoral Researcher at the Department of Signals and Systems, Chalmers University of Technology. From 2006 to 2010, he was a Principal Design Engineer with Quantenna Communications, Inc., Fremont, CA, USA. He is currently Professor and Head of Communication Systems Group, Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. His current research interests include algorithms, resource allocation, synchronization, antenna concepts, and protocol design for vehicular communication systems, as well as different applications of coding.

Paolo Falcone received his Ph.D. degree in Information Technology in 2007 from the University of Sannio, in Benevento, Italy. He is a Professor at the Department of Electrical Engineering, Chalmers University of Technology, Sweden. His research focuses on constrained optimal control, applied to autonomous and semi-autonomous mobile systems, cooperative driving and intelligent vehicles. He is involved in several projects, in cooperation with industry, focusing on autonomous driving, cooperative driving and vehicle dynamics control.

Henk Wymeersch (S'99, M'05) is a Professor with the Department of Electrical Engineering at Chalmers University of Technology, Sweden. Prior to joining Chalmers, he was a postdoctoral researcher from 2005 until 2009 with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology. Henk Wymeersch obtained the Ph.D. degree in Electrical Engineering/Applied sciences in 2005 from Ghent University, Belgium. He served as Associate Editor for IEEE Communication Letters (2009-2013), IEEE Transactions on Wireless Communications (since 2013), and IEEE Transactions on Communications (since 2016). He is the author of Iterative Receiver Design (Cambridge University Press, 2007). His current research interests include cooperative systems and intelligent transportation.